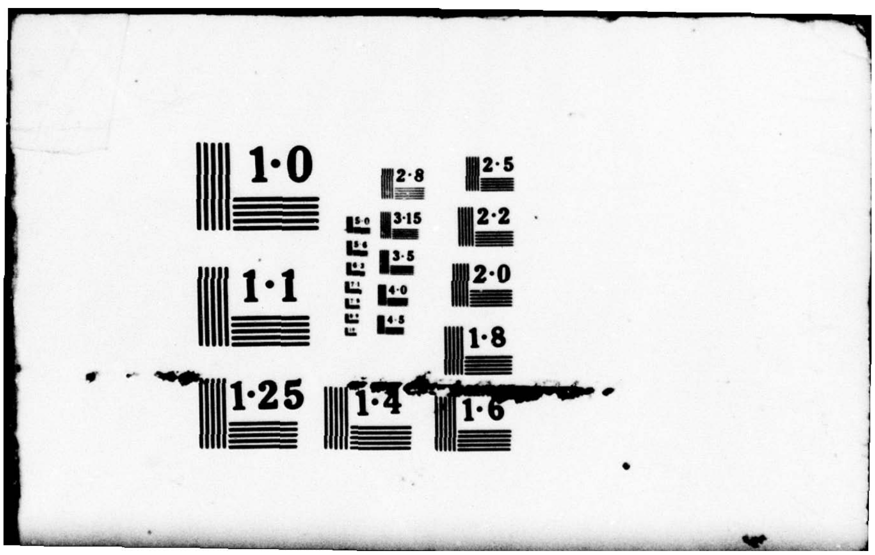
AD-A067 712 SOUTHERN METHODIST UNIV DALLAS TX DEPT OF OPERATIONS--ETC F/G 12/1
SPLITTING THE BUMP IN AN ELIMINATION FACTORIZATION.(U)
MAR 79 R V HELGASON, J K KENNINGTON AFOSR-77-3151
UNCLASSIFIED OREM-79004 AFOSR-TR-79-0446 NL

1 OF 1
ADA
067712

END
DATE
FILMED
6-79
DDC


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
4·5
1·1
1·8
1·25
1·4
1·6

AFOSR-TR- 79-0446

LEVEL II

4

AD A067712

Technical Report OREM 79004

DDC
RECEIVED
APR 17 1979
C

SPLITTING THE BUMP IN AN ELIMINATION FACTORIZATION

R. V. Helgason

J. L. Kennington

DDC FILE COPY

Department of Operations Research
and
Engineering Management

March 1979

Comments and criticisms from interested readers are cordially invited.

ABSTRACT

This exposition presents a method for incorporating a technique known as "splitting the bump" within an elimination form reinversion algorithm. This procedure is designed to reduce fill-in during reinversion and should improve the efficiency of linear programming systems which already use the superior elimination form of the inverse.

ACKNOWLEDGEMENT

This research was supported in part by the Air Force Office of Scientific Research under Grant Number AFOSR 77-3151.

I. INTRODUCTION

Current production linear programming systems are designed to handle problems with 8000 to 16,000 rows and 1000 row problems are considered to be medium sized (see [10, 17]. Fortunately real problems tend to be sparse (the density of the constraint matrix is often less than 1% - see [1, 2, 9, 14, 15]). Hence, a basis for a 1000 row problem may have only (1000) (1000) (1%) = 10,000 nonzero entries. However, the inverse (which is required for the revised simplex algorithm) may be quite dense having almost a million nonzero elements. Consequently, one of the most important design considerations for a computer implementation of the revised simplex algorithm for general linear programs is the technique used to maintain and update the basis inverse.

In order to minimize the storage required to implement this algorithm, production linear programming systems maintain the inverse of the basis in either product or elimination form (see [1, 3, 8, 12, 17]). Computationally the inverse is stored as a sequence of vectors, known as *eta vectors*, and the complete list of the eta's is called the *ETA File*. Each basis change (pivot) results in appending at least one eta vector to the ETA File. Since both the time per pivot and numerical error increase as the length of the ETA File increases, it is necessary to periodically obtain a new factorization of the basis inverse. This process of obtaining a new factorization is called *reinversion*. The objective of a reinversion algorithm is to obtain a factored inverse (i.e., ETA File) in which the sparsity characteristics of the original basis are preserved as much as possible.

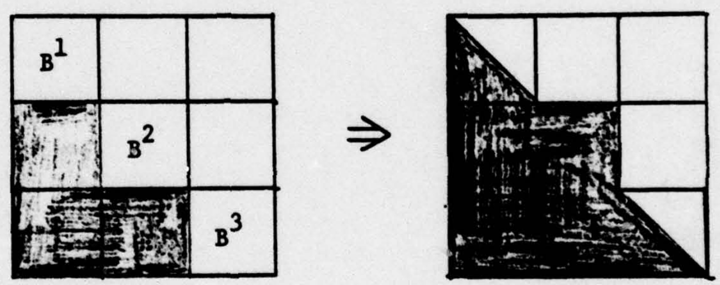
The simplest reinversion technique for a given m-column basis can be thought of as successively reducing the basis to an identity matrix via m pivot operations. The matrices which accomplish this reduction constitute the ETA File, often refered to as a product form factorization. Out of this simple approach, reinversion techniques have evolved which attempt to obtain a sparse factorization by selection of pivot positions, involving a reordering of the columns of the basis and a conceptual reordering of the rows. The row and column pivot ordering employed in reinversion is usually called the *pivot agenda*. Markowitz [11] pioneered the work in this area. Another technique (see [8] and [13]) which has been used in conjunction with the reinversion algorithms is a technique known as *splitting the bump*. All these ideas were directed toward developing a sparse representation using the product form of the inverse. Recently, however, computational evidence indicates that the elimination form factorization is superior to the product form factorization (with respect to both storage requirements and speed (see [2]). The Hellerman-Rarick algorithm can be easily modified to accommodate an elimination form reinversion, but it is not obvious how one may implement the technique known as "splitting the bump" and still make it possible to maintain the elimination form of the inverse during later simplex iterations. The objective of this exposition is to present a reinversion algorithm in which one may achieve some of the benefits of the "splitting the bump technique" when using the elimination form of the inverse.

For this exposition, the i^{th} column of the m x m matrix B is denoted by B(i). The symbol e^i denotes the m-component column vector having i^{th} component one and all other components zero. The symbol η denotes an m-component column vector and η_i denotes the i^{th} component of this vector.

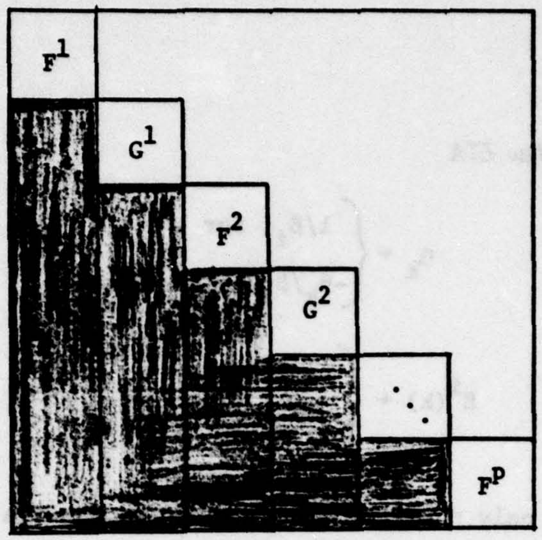
II. FACTORIZATION ALGORITHMS

Let B be any m by m nonsingular matrix. In this section we present two algorithms for obtaining a factorization of B^{-1}. The first algorithm produces a product form factorization which corresponds to the method for solving a system of linear equations known as Gauss-Jordan reduction while the second algorithm produces an elimination form factorization which corresponds to a Gauss reduction (see [4]).

By row and column interchanges, B may be placed in the following form:



where B^1 and B^3 are lower triangular matrices with nonzeroes on their diagonals. We assume that if B^2 is nonvacuous, every row and column has at least two nonzero entries, so that no rearrangement of B^2 can expand the size of B^1 or B^3. B^2 is called the *bump*, *merit*, or *heart section*. We further require the heart section to assume the following form:



where G^k's are either vacuous or lower triangular with nonzeroes on the diagonal. The only partitions in B having columns with nonzeroes above the diagonal are the F^k's which are called *external bumps*. The columns extending above the diagonal are called *spikes* or *spike columns*. An external bump is characterized as follows:

(i) the last column of an external bump will be a spike with a nonzero lying in the topmost row of the external bump, and

(ii) the nonspike columns have nonzero diagonal elements.

The algorithms of Hellerman and Rarick [6, 7] produce such a structure for any nonsingular matrix B, and we shall call a matrix having this structure an HR matrix.

The product form factorization algorithm for an HR matrix is given as follows:

ALG 1: PRODUCT FORM FACTORIZATION FOR A HR MATRIX

0. *Initialization*

Set $i \leftarrow 1$ and $\beta \leftarrow B(1)$.

1. *Obtain New ETA*

Set $$\eta_k \leftarrow \begin{cases} 1/\beta_i, & \text{for } k = i \\ -\beta_k/\beta_i, & \text{otherwise,} \end{cases}$$

and set $$E^i(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise.} \end{cases}$$

(Note: only η and i need be saved in the ETA File)

2. *Test For Termination*

If $i = m$, terminate, otherwise, $i \leftarrow i + 1$.

3. *Test For Spike*

If $B(i)$ is not a spike, set $\beta \leftarrow B(i)$ and go to 1.

4. *Spike Update*

Let $B(k)$ correspond to the first column of the external bump containing $B(i)$. Set $\beta \leftarrow E^{i-1} \cdots E^k B(i)$.

5. *Swap Spikes If Pivot Element Equals Zero*

If $\beta_i \neq 0$, go to 1; otherwise, there is some spike $B(j)$ in the same external bump having $j > i$ such that the i^{th} component of $E^{i-1} \cdots E^k B(j)$ is nonzero. Set $\beta \leftarrow E^{i-1} \cdots E^k B(j)$, interchange $B(i)$ and $B(j)$ and go to 1.

In practice, the test for a zero pivot element in step 5 is usually replaced by a tolerance test. Let TOL denote the tolerance to be used in the test. If $|\beta_i| \leq$ TOL, then β_i is treated as if it is zero. Discussions of tolerences may be found in [12, 15, 16]. Similar tolerance tests would ordinarily be incorporated in the other algorithms to be presented in this exposition. To simplify the presentation they have been omitted here.

Justification for ALG 1 is given in [5]. At termination, $B^{-1} = E^m E^{m-1} \ldots E^1$. Furthermore, we see that for nonspike columns, each β and consequently each η has exactly the same number of nonzeroes as the corresponding column of B. However, the η for a spike column may have a higher density than the original column of B. The phenomena of an η having a higher density than the corresponding column of B is known as *fill-in*.

It is well known that fill-in may be reduced by applying an elimination form factorization rather than the product form (see [1]). The elimination form algorithm for an HR matrix is given as follows:

ALG 2: ELIMINATION FORM FACTORIZATION FOR A HR MATRIX

0. *Initialization*

Set $i \leftarrow 1$, $r \leftarrow 0$, and $\beta \leftarrow B(1)$.

1. *Obtain New Lower Eta*

Set $$\eta_k \leftarrow \begin{cases} 1/\beta_i, \text{ for } k = i \\ -\beta_k/\beta_i, \text{ for } k > i \\ 0, \text{ otherwise}, \end{cases}$$

and set $$L^i(k) \leftarrow \begin{cases} \eta, \text{ for } k = i \\ e^k, \text{ otherwise}. \end{cases}$$

(Note: only η and i need be saved in the Eta File for the lower factors)

2. *Test For Termination*

If $i = m$, terminate; otherwise, $i \leftarrow i + 1$.

3. *Test For Spike*

If $B(i)$ is not a spike, set $\beta \leftarrow B(i)$ and go to 1.

4. *Spike Update*

Let $B(k)$ correspond to the first column of the external bump containing $B(i)$. Set $\beta \leftarrow L^{i-1} \cdots L^{k} B(i)$.

5. *Swap Spikes If Pivot Element Equals Zero*

If $\beta_i \neq 0$, go to 6; otherwise, there is some spike $B(j)$ in the same external bump having $j > i$ such that the i^{th} component of $L^{i-1} \cdots L^{k} B(j)$ is nonzero. Set $\beta \leftarrow L^{i-1} \cdots L^{k} B(j)$ and interchange $B(i)$ and $B(j)$.

6. *Obtain New Upper Eta*

Set $r \leftarrow r + 1$

$$\text{Set} \qquad \eta_k \leftarrow \begin{cases} 1, & \text{for } k = i \\ -\beta_k, & \text{for } k < i \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{set} \qquad U^r(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise,} \end{cases}$$

and go to 1.

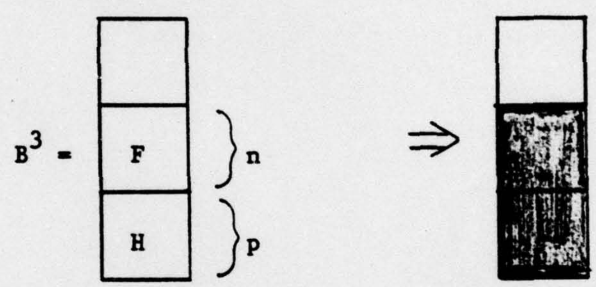
(Note: only η and i need be saved in the Eta File for the upper factors)

At the termination of ALG 2, $B^{-1} = U^1 \cdots U^r L^m \cdots L^1$, where the upper eta's are upper triangular and the lower eta's are lower triangular. As with ALG 1 fill-in has been restricted to the eta's corresponding to spike columns.

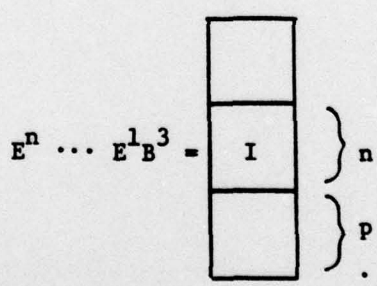
III. SPLITTING THE BUMP

In an attempt to reduce the fill-in which occurs in spike columns during a reinversion using ALG 1, a variation of the elimination form factorization algorithm (attributed to Martin Beale, see [13]) has been used by some practitioners. This technique has been called "splitting the bump" after its treatment of external bump columns.

Consider a set of columns of the basis corresponding to an external bump, say



where F is an external bump. Suppose F contains q spikes. If we apply the standard product form algorithm, we obtain a sets of eta's such that



The eta's corresponding to the q spikes may incur fill-in. If we split the bump, the fill-in can be restricted to the n rows corresponding to F; hence, we may avoid fill-in in the last p rows. Since p may be much larger than n, the savings could be substantial. The price which must be paid to attain this reduction in fill-in is that each external bump will require 2n + q eta's rather than n. The savings in fill-in is typically so great that it offsets the additional storage which must be given up for the additional eta's.

We first develop a set of n eta's such that

$$E^n \cdots E^1 B^3 = \begin{array}{|c|l} \hline \\ \hline F^1 & \left.\right\} n \\ \hline H & \left.\right\} p \\ \hline \end{array} \qquad \Rightarrow$$

where F^1 has +1's on the diagonal and zeroes below the diagonal. The only columns having nonzeroes above the diagonal are spike columns. Next we zero out the spikes in F^1 by developing q eta's such that

$$E^{n+q} \cdots E^n \cdots E^1 B^3 = \begin{array}{|c|} \hline \\ \hline I \\ \hline H \\ \hline \end{array} \; .$$

Finally, we zero out the H via the addition of n more eta's. That is,

$$E^{2n+q} \ldots E^{n+q} \ldots E^1 B^3 = \begin{array}{|c|} \hline \\ \hline I \\ \hline \\ \hline \end{array} \; .$$

The product form algorithm incorporating the "splitting the bump" technique is given as follows:

ALG 3: PRODUCT FORM FACTORIZATION FOR A HR MATRIX INCLUDING SPLITTING THE BUMP

0. *Initialization*

Set $i \leftarrow 1$, $j \leftarrow 1$, and $\beta \leftarrow B(1)$. If $B(1)$ is in an external bump, to to 4.

1. *Obtain New Lower Eta*

Set $$\eta_k \leftarrow \begin{cases} 1/\beta_i\text{, for } k = i \\ -\beta_k/\beta_i\text{, otherwise,} \end{cases}$$

and set $$E^j(k) \leftarrow \begin{cases} \eta\text{, for } k = i \\ e^k\text{, otherwise.} \end{cases}$$

Set $j \leftarrow j + 1$.

2. *Test For Termination*

If $i = m$, terminate; otherwise, $i \leftarrow i + 1$ and $\beta \leftarrow B(i)$.

3. *Test For External Bump*

If $B(i)$ is not the first column in an external bump, go to 1.

4. *Initialization For Bump*

Set $$\begin{cases} \ell \leftarrow j \\ s \leftarrow i \\ b \leftarrow \text{number of columns in this external bump} \\ t \leftarrow i + b - 1 \end{cases}$$

5. *Obtain Lower Eta*

$$\text{Set} \qquad \eta_k \leftarrow \begin{cases} 1/\beta_i, \text{ for } k = i \\ -\beta_k/\beta_i, \text{ for } i < k \leq t \\ 0, \text{ otherwise,} \end{cases}$$

$$\text{and set} \qquad E^j(k) \leftarrow \begin{cases} \eta, \text{ for } k = i \\ e^k, \text{ otherwise.} \end{cases}$$

Set $j \leftarrow j + 1$.

6. *Test For End Of Bump*

If $i = t$, go to 10; otherwise, $i \leftarrow i + 1$.

7. *Test For Spike*

If $B(i)$ is not a spike, set $\beta \leftarrow B(i)$ and go to 5.

8. *Spike Update*

Set $\beta \leftarrow E^{j-1} \cdots E^{\ell}B(i)$.

9. *Swap Spikes If Pivot Element Equals Zero*

If $\beta_i \neq 0$, go to 5; otherwise, there is some spike $B(r)$ in the same external bump having $r > i$ such that the i^{th} component of $E^{j-1} \cdots E^{\ell}B(r)$ is nonzero. Set $\beta \leftarrow E^{j-1} \cdots E^{\ell}B(r)$, interchange $B(r)$ and $B(i)$ and go to 5.

10. *Obtain Upper Eta*

$$\text{Set} \qquad \eta_k \leftarrow \begin{cases} 1, \text{ for } k = i \\ -\beta_k, \text{ for } k < i \\ 0, \text{ otherwise,} \end{cases}$$

and set $E^j(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise.} \end{cases}$

Set $j \leftarrow j + 1$ and $i \leftarrow i - 1$.

11. *Test For Beginning Of Bump*

If $i = s$, set $\beta \leftarrow B(i)$ and go to 13.

12. *Test For Spike*

If $B(i)$ is not a spike, set $i \leftarrow i - 1$ and go to 11; otherwise, set $\beta \leftarrow E^{j-1} \cdots E^{\ell} B(i)$ and go to 10.

13. *Obtain Lower Eta*

Set $\eta_k \leftarrow \begin{cases} 1, & \text{for } k = i \\ -\beta_k, & \text{for } k > t \\ 0, & \text{otherwise,} \end{cases}$

and set $E^j(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise.} \end{cases}$

Set $j \leftarrow j + 1$.

14. *Test For End Of Bump*

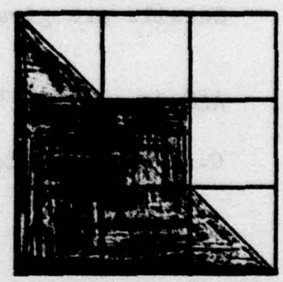
If $i = t$, go to 2; otherwise $i \leftarrow i + 1$, set $\beta \leftarrow B(i)$, and go to 13.

At the termination of ALG 3, $B^{-1} = E^{j-1} \cdots E^{1}$. Furthermore, all eta's are either upper triangular or lower triangular, but they are intermixed.

Recall that ALG 2 produces a set of lower triangular factors followed by a set of upper triangular factors. If one incorporates the "splitting the bump" technique into ALG 2 and apply it to an HR matrix, the lower and upper factors become intermingled. Once the factors have become intermingled, we may no longer use the important algorithm of Forrest and Tomlin [2] to maintain the elimination form. We now show how one may achieve a partial "splitting of the bump" while simultaneously maintaining a partitioning of the upper and lower factors.

Recall that an HR matrix takes the following form:

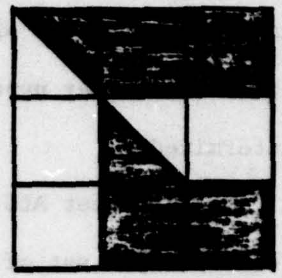
$$B = \begin{array}{|c|c|c|} \hline B^1 & & \\ \hline B^4 & B^2 & \\ \hline B^6 & B^5 & B^3 \\ \hline \end{array} \Rightarrow$$



.

ALG 3 (Product Factorization With Splitting The Bump) eliminates all fill-in in B^5. By a rearrangement of rows and columns, B may be placed in the following form:

$$\bar{B} = \begin{array}{ccc} \left[\begin{array}{c|c|c} \bar{B}^3 & \bar{B}^6 & \bar{B}^5 \\ \hline & B^1 & \\ \hline & B^4 & B^2 \end{array}\right] \\ \underbrace{\qquad}_{u} \quad \underbrace{\qquad}_{v} \quad \underbrace{\qquad}_{w} \end{array} \Rightarrow$$



where $\bar{B}^6$ and $\bar{B}^5$ are row permutations of B^6 and B^5, respectively, and $\bar{B}^3$ is a row and column permutation of B^3. Applying a variation of ALG 2 to $\bar{B}$ eliminates all fill-in in $\bar{B}^5$ while representing $\bar{B}^{-1}$ as a product of upper factors followed by a product of lower factors. The details are given in the following algorithm.

ALG 4: ELIMINATION FORM FACTORIZATION FOR A MODIFIED HR MATRIX

0. *Initialization*

Set $i \leftarrow u + 1$ and $\beta \leftarrow \bar{B}(i)$. If $v = 0$, go to 3.

1. *Obtain New Lower Eta*

$$\text{Set} \quad \eta_k \leftarrow \begin{cases} 1/\beta_i, \text{ for } k = i \\ -\beta_k/\beta_i, \text{ for } k > i \\ 0, \text{ otherwise}, \end{cases}$$

$$\text{and set} \quad L^i(k) \leftarrow \begin{cases} \eta, \text{ for } k = i \\ e^k, \text{ otherwise}. \end{cases}$$

2. *Test For End Of Section 2*

Set $i \leftarrow i + 1$ and $\beta \leftarrow \bar{B}(i)$. If $i \leq u + v$, go to 1. If $w = 0$, to to 8.

3. *Obtain New Lower Eta*

$$\text{Set} \quad \eta_k \leftarrow \begin{cases} 1/\beta_i, \text{ for } k = i \\ -\beta_k/\beta_i, \text{ for } k > i \\ 0, \text{ otherwise}, \end{cases}$$

$$\text{and set} \quad L^i(k) \leftarrow \begin{cases} \eta, \text{ for } k = i \\ e^k, \text{ otherwise}. \end{cases}$$

4. *Test For End Of Section 3*

If $i = m$, go to 8; otherwise, $i \leftarrow i + 1$.

5. *Test For Spike*

If $\bar{B}(i)$ is not a spike, set $\beta \leftarrow \bar{B}(i)$ and go to 3.

6. *Spike Update*

Set $\beta \leftarrow L^{i-1} \cdots L^{u+v+1}\bar{B}(i)$.

7. *Swap Spikes If Pivot Element Equals Zero*

If $\beta_i \neq 0$, go to 3; otherwise, there is some spike $B(j)$ in the same external bump having $j > i$ such that the i^{th} component of $L^{i-1} \cdots L^{u+v+1}\bar{B}(j)$ is nonzero. Set $\beta \leftarrow L^{i-1} \cdots L^{u+v+1}\bar{B}(j)$, interchange $\bar{B}(i)$ and $\bar{B}(j)$ and go to 3.

8. *Obtain New Upper Eta*

$$\text{Set} \qquad \eta_k \leftarrow \begin{cases} 1, & \text{for } k = i \\ -\beta_k, & \text{for } k < i \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{and set} \qquad U^i(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise.} \end{cases}$$

9. *Check For End Of Section 2*

Set $i \leftarrow i - 1$. If $i = 0$, terminate.

If $i = u$, set $\beta \leftarrow B(i)$ and go to 12.

10. *Set Column*

If $i > u + r$, set $\beta \leftarrow L^i \cdots L^{u+v+1}\bar{B}(i)$; otherwise,

set $\beta \leftarrow \bar{B}(i)$. Go to 8.

11. *Obtain New Upper Eta*

$$\text{Set} \qquad \eta_k \leftarrow \begin{cases} 1/\beta_i, & \text{for } k = i \\ -\beta_k/\beta_i, & \text{for } k < i \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{and set} \qquad U^i(k) \leftarrow \begin{cases} \eta, & \text{for } k = i \\ e^k, & \text{otherwise.} \end{cases}$$

12. *Check For Termination*

If $i = 1$, terminate; otherwise $\beta \leftarrow \bar{B}(i)$ and go to 11.

79 04 13 050

At the termination of ALG 4, $\bar{\bar{B}}^{-1} = U^1 \cdots U^m L^m \cdots L^{u+1}$, and the fill-in has been restricted to B^2. Hence, ALG 4 gains some of the benefits of the "splitting the bump" technique while maintaining a partitioning of the upper and lower factors. The benefits are not as great as with ordinary "bump splitting" since each individual external bump is split, whereas here the split is with respect to the entire heart section.

A variation of ALG 4 has also been used by Tomlin [15]. Our contribution is that we have tied together the ideas of "splitting the bump" in both the product and elimination form factorizations and we have explicitly indicated via ALG's 3 and 4 how these may be implemented.

REFERENCES

1. Brayton, R. K., F. G. Gustavson, and R. A. Willoughby, "Some Results on Sparse Matrices," Mathematics of Computation, 24, 112, (1970), 937-954.
2. Forrest, J. J. H., and J. A. Tomlin, "Updated Triangular Factors of the Basis to Maintain Sparsity in the Product From Simplex Method," Mathematical Programming, 2, (1972), 263-278.
3. Greenberg, H. J., and D. C. Rarick, "Determining GUB Sets Via An Invert Agenda Algorithm," Mathematical Programming, 7, (1974), 240-244.
4. Hadley, G., Linear Algebra, Addison Wesley Publishing Co., Inc., Reading, Massachusetts, (1964).
5. Helgason, R. V., and J. L. Kennington, "Spike Swapping in Basis Reinversion," Technical Report OREM 79001, Department of Operations Research, Southern Methodist University, Dallas, Texas, (1979).
6. Hellerman, E. and D. Rarick, "Reinversion With the Preassigned Pivot Procedure," Mathematical Programming, 1, (1971), 195-216.
7. Hellerman, E., and D. Rarick, "The Partitioned Preassigned Pivot Procedure (P^4)," Sparse Matrices and Their Applications, Edited by D. Rose and R. Willoughby, Plenum Press, New York, New York, (1972).
8. Kalan, J. E., "Aspects of Large-Scale In-Core Linear Programming," Proceedings of the ACM, (1971), 304-313.
9. Luhks, R. A., "Large-Scale Linear Programming Problems: Analysis and Generation," unpublished dissertation, Department of Computer Science and Operations Research, Southern Methodist University, (1972).

10. Marceau, I. W., R. J. Lermit, J. C. McMillan, T. Yamamoto, and S. S. Yardley, "Linear Programming on Illiac IV," Proceedings of Fourth Australian Computer Conference, Adelaide, Australia, 481-485, (1969).

11. Markowitz, H. M., "The Elimination From of the Inverse and Its Application to Linear Programming," Management Science, 3, (1975), 255-269.

12. Orchard-Hays, W., Advanced Linear-Programming Computing Techniques, McGraw-Hill, New York, New York, (1968).

13. Orchard-Hays, W., "An Exercise in Basis Inversion," SIGMAP Newsletter, No. 15, (1973).

14. Tomlin, J. A., "Modifying Triangular Factors of the Basis in the Simplex Method," in Sparse Matrices and Their Applications, Edited by D. J. Rose and R. A. Willoughby, Plenum Press, New York, New York, (1972).

15. Tomlin, J. A., "Pivoting for Size and Sparsity in Linear Programming Inversion Routines," Journal of the Institute of Mathematics and its Applications, 10, (1972), 289-295.

16. Tomlin, J. A., "An Accuracy Test for Updating Triangular Factors," Mathematical Programming Study 4, (1975), 142-145.

17. White, W. W., "A Status Report on Computing Algorithms for Mathematical Programming," Computing Surveys, 5, 3, (1973), 135-166.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 AFOSR TR-79-0446	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Splitting the Bump in an Elimination Factorization.		5. TYPE OF REPORT & PERIOD COVERED 9 Interim rept.,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 R. V. Helgason J. K. Kennington		8. CONTRACT OR GRANT NUMBER(s) 15 AFOSR-77-3151
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research Southern Methodist University Dallas, Texas 75275		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 2304/46 17 61102F
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM Bolling AFB, D.C. 20332	11 Mar 79	12. REPORT DATE 3/79
		13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 24p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release; its distribution is unlimited.

14 OREM-79004

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear Programming, Reinversion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This exposition presents a method for incorporating a technique known as "splitting the bump" within an elimination form reinversion algorithm. This procedure is designed to reduce fill-in during reinversion and should improve the efficiency of linear programming systems which already use the superior elimination form of the inverse.

DD FORM 1 JAN 73 1473

410 597

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 04 13 050